Claims 1—23: Cancelled.

24. (Currently amended.) A method for determining shadowing cast onto a bump mapped surface by one or more bumps thereon, where the bump mapped surface is represented by a plurality of polygons each having a plurality of vertices, each vertex defining a point in a tangent plane of the bump mapped surface, where a normal vector at each said vertex is perturbed to define a bump map of the bump mapped surface that includes a plurality of bumps on the surface, the method comprising:

defining a horizon map of the surface including, for each of a plurality of radial directions in the tangent plane around each said vertex, the largest angle between the normal vector and any direct ray of light to the vertex; ~~and~~

storing the horizon map as a set of texture maps corresponding to the surface on the three-dimensional graphical ~~object.~~ object; and

wherein for each said vertex, each said texture map has a plurality of color channels, each said color channel having encoded therein the largest angle between the normal vector and any direct ray of light to the vertex that does not contact any of the bumps on the surface.

25. (Original.) The method as defined in Claim 24, further comprising:

executing an interactive program that defines a light source by a direction relative to:

the tangent plane of the surface; and

the normal of each said vertex;

determining at least two of the radial directions stored in the horizon map with respect to the direction of the light source;

interpolating from the largest angles at the at least two radial directions of the horizon map to obtain the portion of each said polygon that is in the light from the light source;

rendering the surface on the three-dimensional graphical object, from a given point of view, to include only the portion of each said polygon that is in the light from the light source;

interactively receiving instructions to change the direction of the light source relative to the surface on the three-dimensional graphical object;

repeating the determining by interpolation step using the changed direction of the light source relative to the surface on the three-dimensional graphical object; and

repeating the rendering.

26. (Cancelled.)

27. (Currently amended.) The method as defined in Claim [[26]]24, wherein:

the set of texture maps comprises two texture maps; and

each said texture map:

~~corresponding~~ corresponds to not more that four of the radial directions; and

has four ~~colors~~ color channels each having not more than one radial direction encoding for the corresponding largest angle for each vertex between the normal vector and any direct ray of light to the vertex that does not contact any of the bumps on the surface.

28. (Cancelled.)

29. (Original.) A method for rendering a frame in a frame buffer represented by a plurality of pixels, the frame depicting a three dimensional graphical object that is defined by surfaces each having a surface geometry defined by a parameterization of a plurality of polygons, each polygon being defined by vertices, each vertex (u,v) having a bump map scalar value F(u,v) defining a bump, each vertex (u,v) having a perturbed normal map vector value N'(u,v) from the bump map scalar value F(u,v), each vertex (u,v) having M basis maps (Bk=1...M (s, t)), each vertex (u,v) having M horizon maps for each of M radial directions (θk=1...M) in the tangent plane of the vertex (u, v), wherein each of the M horizon maps includes the largest angle ϕ(u,v,θk=1...M) between the perturbed normal map vector value N'(u, v) and any direct ray to the vertex (u, v), the method comprising:

(i) defining a light vector (L) having a direction defined by:

the angle θ_L between the light vector (L) and the plane formed by coordinates (u,v); and

the angle ϕ_L between the light vector (L) and the perturbed normal map vector value N'(u,v);

(ii) inverting a local tangent frame by $[P_u , P_v ,N]^{-1} = [S^T, T^T, N^T]$ to obtain an inverted tangent frame (S,T);

(iii) taking the dot product of the light vector(L):

with the first two components of the inverted frame (S,T) to obtain the projection of the light vector (L) into the

coordinate space in the tangent plane at a coordinate pair (s,t); and

with the perturbed normal map vector value N'(u, v) at the vertex to obtain cosine (ϕ_L);

(iv) setting the camera angle of the rendering to a point of view to be rendered for the frame to write into the space of a texture map;

(v) computing first and second texture effects on each pixel in the frame buffer, respectively, from first and second sets of the M directions stored as texture maps, wherein each said largest angle $\phi(u,v,\theta_k)$ is in one of a plurality of color channels of a texture map, wherein one of the M directions corresponds to each said color channel of the texture map, and wherein the first and second texture effects are defined, respectively by:

a basis map B_1 (s, t) of said M basis maps ($B_{k=1...M}$ (s, t)), and a horizon map, ϕ_1 (u, v) of said M horizon maps; and

a basis map B_2 (s, t) of said M basis maps ($B_{k=1...M}$ (s, t)), and a horizon map, ϕ_2 (u, v) of said M horizon maps;

(vi) storing the first and second texture effects for each pixel in the frame buffer in a texture map $\phi(\theta_{LIGHT})$;

(vii) for a given camera perspective view:

on a first operation, rendering a model of the three dimensional graphical object with an ambient low level lighting term only;

on a second operation:

setting an alpha test and a stencil function to accept only those pixels in the frame buffer that have a non-zero alpha test result;

setting a color mask upon the frame buffer, wherein the color mask will not write to the color channels of the texture maps, whereby the ambient low level lighting term previously rendered is preserved;

drawing, using multi-texturing, for the angle (ϕ_L) and the angle to the horizon (θ_L), the first and second texture effects on each pixel in the frame buffer, which are, respectively, cosine(ϕ_L) that is greater than $\phi_{L,}$ and $\phi(\theta_{LIGHT})$;

on a third operation:

setting the alpha test off;

setting the color mask upon the frame buffer to allow writing to the color channels of the texture maps;

setting the stencil function to only draw pixels in the frame buffer that have a non-zero alpha test result;

setting a blending function to accumulate into the frame buffer with the ambient low level lighting term;

drawing the vector valued perturbed normal map N'(u,v); and

displaying the three dimensional graphical object defined by pixels in the frame buffer, wherein the displayed object has an ambient only term in shadowed regions and normal bump mapping in non-shadowed regions.

30. (Original.) The method as defined in Claim 29, further comprising:

interactively receiving instructions that change the definition of the light vector (L) with respect to each vertex (u,v); and

repeating (i) through (vii) using the changed definition of the light vector (L).

Claims 31—41: Cancelled.

42. (New.) One or more computer-readable media comprising computer-executable instructions for determining shadowing cast onto a bump mapped surface by one or more bumps thereon, where the bump mapped surface is represented by a plurality of polygons each having a plurality of vertices, each vertex defining a point in a tangent plane of the bump mapped surface, where a normal vector at each said vertex is perturbed to define a bump map of the bump mapped surface that includes a plurality of bumps on the surface, the computer-executable instructions comprising instructions for:

defining a horizon map of the surface including, for each of a plurality of radial directions in the tangent plane around each said vertex, the largest angle between the normal vector and any direct ray of light to the vertex;

storing the horizon map as a set of texture maps corresponding to the surface on the three-dimensional graphical object; and

wherein for each said vertex, each said texture map has a plurality of color channels, each said color channel having encoded therein the largest angle between the normal vector and any direct ray of light to the vertex that does not contact any of the bumps on the surface.

43. (New.) The one or more media as recited in claim 42, comprising further instructions for:

executing an interactive program that defines a light source by a direction relative to:

the tangent plane of the surface; and

the normal of each said vertex;

determining at least two of the radial directions stored in the horizon map with respect to the direction of the light source;

interpolating from the largest angles at the at least two radial directions of the horizon map to obtain the portion of each said polygon that is in the light from the light source;

rendering the surface on the three-dimensional graphical object, from a given point of view, to include only the portion of each said polygon that is in the light from the light source;

interactively receiving instructions to change the direction of the light source relative to the surface on the three-dimensional graphical object;

repeating the determining by interpolation step using the changed direction of the light source relative to the surface on the three-dimensional graphical object; and

repeating the rendering.

44. (New.) The one or more media as recited in claim 42, wherein:

the set of texture maps comprises two texture maps; and

each said texture map:

corresponds to not more that four of the radial directions; and

has four color channels each having not more than one radial direction encoding for the corresponding largest angle for each vertex between the normal vector and any direct ray of light to the vertex that does not contact any of the bumps on the surface.

45. (New.) One or more computer-readable media comprising computer-executable instructions for rendering a frame in a frame buffer represented by a plurality of pixels, the frame depicting a three dimensional graphical object that is defined by surfaces each having a surface geometry defined by a parameterization of a plurality of polygons, each polygon being defined by vertices, each vertex (u,v) having a bump map scalar value F(u,v) defining a bump, each vertex (u,v) having a perturbed normal map vector value N'(u,v) from the bump map scalar value F(u,v), each vertex (u,v) having M basis maps (Bk=1...M (s, t)), each vertex (u,v) having M horizon maps for each of M radial directions (θk=1...M) in the tangent plane of the vertex (u, v), wherein each of the M horizon maps includes the largest angle ϕ(u,v,θk=1...M) between the perturbed normal map vector value N'(u, v) and any direct ray to the vertex (u, v), the computer-executable instructions comprising instructions for:

(i) defining a light vector (L) having a direction defined by:

the angle θ_L between the light vector (L) and the plane formed by coordinates (u,v); and

the angle ϕ_L between the light vector (L) and the perturbed normal map vector value N'(u,v);

(ii) inverting a local tangent frame by $[P_u , P_v ,N]^{-1} = [S^T, T^T, N^T]$ to obtain an inverted tangent frame (S,T);

(iii) taking the dot product of the light vector(L):

with the first two components of the inverted frame (S,T) to obtain the projection of the light vector (L) into the coordinate space in the tangent plane at a coordinate pair (s,t); and

with the perturbed normal map vector value N'(u, v) at the vertex to obtain cosine (ϕ_L);

(iv) setting the camera angle of the rendering to a point of view to be rendered for the frame to write into the space of a texture map;

(v) computing first and second texture effects on each pixel in the frame buffer, respectively, from first and second sets of the M directions stored as texture maps, wherein each said largest angle $\phi(u,v,\theta_k)$ is in one of a plurality of color channels of a texture map, wherein one of the M directions corresponds to each said color channel of the texture map, and wherein the first and second texture effects are defined, respectively by:

a basis map B_1 (s, t) of said M basis maps ($B_{k=1...M}$ (s, t)), and a horizon map, ϕ_1 (u, v) of said M horizon maps; and

a basis map $B_2(s, t)$ of said M basis maps $(B_{k-1...M}(s, t))$, and a horizon map, $\phi_2(u, v)$ of said M horizon maps;

(vi) storing the first and second texture effects for each pixel in the frame buffer in a texture map $\phi(\theta_{LIGHT})$;

(vii) for a given camera perspective view:

on a first operation, rendering a model of the three dimensional graphical object with an ambient low level lighting term only;

on a second operation:

setting an alpha test and a stencil function to accept only those pixels in the frame buffer that have a non-zero alpha test result;

setting a color mask upon the frame buffer, wherein the color mask will not write to the color channels of the texture maps, whereby the ambient low level lighting term previously rendered is preserved;

drawing, using multi-texturing, for the angle (ϕ_L) and the angle to the horizon (θ_L), the first and second texture effects on each pixel in the frame buffer, which are, respectively, cosine(ϕ_L) that is greater than ϕ_L, and $\phi(\theta_{LIGHT})$;

on a third operation:

setting the alpha test off;

setting the color mask upon the frame buffer to allow writing to the color channels of the texture maps;

setting the stencil function to only draw pixels in the frame buffer that have a non-zero alpha test result;

setting a blending function to accumulate into the frame buffer with the ambient low level lighting term;

drawing the vector valued perturbed normal map N'(u,v); and

displaying the three dimensional graphical object defined by pixels in the frame buffer, wherein the displayed object has an ambient only term in shadowed regions and normal bump mapping in non-shadowed regions.

46. (New.) The one or more media as recited in claim 45, comprising further instructions for:

interactively receiving instructions that change the definition of the light vector (L) with respect to each vertex (u,v); and

repeating (i) through (vii) using the changed definition of the light vector (L).